

ERLANGMS: Uma Plataforma em Erlang/OTP para Modernização de Sistemas Legados através de uma Abordagem Orientada a Serviços na UnB

Everton Agilar¹, Alysso Ribeiro¹, Renato Ribeiro¹, Eduardo Teixeira¹,

¹Universidade de Brasília – Campus Universitário Darcy Ribeiro
Caixa Postal – 70910-90 – Brasília – DF – Brasil

{evertonagilar, alyssonribeiro, rcarauta, eduhennr}@unb.br

Resumo. Nos últimos anos, a modernização dos sistemas legados da Universidade de Brasília (UnB) tem sido prioridade para o CPD/UnB. A Arquitetura Orientada a Serviços (SOA) surge como uma maneira de solucionar este problema, disponibilizando uma abstração de alto nível entre as aplicações e a camada de serviço. Este artigo aborda este tema e descreve alguns resultados obtidos com o uso da plataforma ERLANGMS desenvolvido sob uma abordagem orientada a serviços que compreende um processo de modernização e um barramento de serviços aderente ao estilo arquitetural Representational State Transfer (REST). Mais especificamente, são discutidos os principais resultados alcançados, as quais destacam-se a definição de um processo de modernização denominado SMSOC para guiar as atividades de migração de sistemas legados e alguns serviços de apoio como um proxy LDAP para unificar o login de usuários e um serviço de autorização OAuth 2 para controlar o acesso aos serviços REST.

1. Introdução

Os sistemas legados correspondem às aplicações que sustentam o funcionamento nacional de uma Instituição e consolidam a maior parte das informações corporativas [Bennett 1995]. Na Universidade de Brasília (UnB), há uma gama considerável de sistemas legados desenvolvido ao longo dos últimos 20 anos pelo CPD/UnB que consistem em um arcabouço de regras de negócios que são de vital importância para o pleno funcionamento da Instituição. Entretanto, com as sucessivas revisões nas regras de negócios para mantê-los alinhados com as necessidades e a obsolescência tecnológica desses sistemas, tornaram-se rígidos e inflexíveis, a ponto de serem de difícil manutenção e evolução.

De forma geral, os sistemas da UnB dividem-se em três áreas de negócio: área acadêmica, administrativa e de pessoal. A maioria desses sistemas foram construídos em diferentes linguagens de programação, arquiteturas e plataformas que não conversam entre si, a não ser, por meio do banco de dados. Durante muitos anos, a linguagem de programação VB foi a predominante. Os dois sistemas mais importantes escritos em VB são o *Sistema Acadêmico (SIGRA)* e o *Sistema de Pessoal (SIPES)*, sendo os demais escritos em VB.Net, C#, PHP, ASP e Java (a plataforma atual).

Neste cenário, as tradicionais práticas de manutenção deixam de atender às organizações, que buscam formas de reduzir os custos com a manutenção, maximizar a integração entre os sistemas, torná-los mais flexíveis às mudanças de forma para prolongar sua vida útil e facilitar a evolução desses sistemas [Bisbal et al. 1999].

Este artigo apresenta a plataforma ERLANGMS desenvolvido pelo CPD/UnB para apoiar a modernização de sistemas legados na UnB. Mais especificamente, o artigo discute os seguintes resultados:

- Desenvolvimento de serviços utilizando o processo de modernização SMSOC
- Autenticação de usuários através um serviço proxy LDAP
- Autenticação e autorização de serviços REST com um serviço OAuth2

2. Método

Com o objetivo de conduzir a modernização dos sistemas legados na UnB, optou-se por experimentar com a arquitetura orientada a serviços, particularmente seguindo o *estilo arquitetural REST*, adotado em muitas Instituições por causa da facilidade de se invocar serviços web a partir de qualquer sistema [Fielding 2000].

Para isso, foi proposta a *plataforma ERLANGMS*, que compreende: (a) um barramento de serviços próprio desenvolvido em *Erlang/OTP* para publicação de serviços REST; (b) um processo de modernização denominado *Software Modernization through Service Oriented Computing (SMSOC)* para guiar os trabalhos de modernização e disponibilizar uma arquitetura de software padronizada para criação dos serviços e; (c) um kit de desenvolvimento (SDK) para desenvolvimento de serviços na linguagem Java ¹.

Embora *SOA* seja um tema de crescente interesse por parte dos pesquisadores e da indústria, identificou-se a necessidade prévia de condução de um mapeamento sistemático para caracterizar a modernização de sistemas legados no contexto da manutenção de software [Agilar et al. 2016]. Desse modo, muitas decisões de design da plataforma foram definidas a partir deste estudo prévio onde verificou-se que a maior parte das contribuições na literatura estão relacionados aos aspectos gerenciais da modernização de software (55,88% das publicações) e há poucos relatos de contribuições que descrevem (ou validem) técnicas ou ferramentas de modernização de software como ERLANGMS.

3. Resultados

Nesta seção são descritos os resultados deste trabalho, A Subseção 3.1 apresenta uma visão geral da plataforma ERLANGMS, a Subseção 3.2 discute as experiências como o uso do processo SMSOC. A Subseção 3.3 descreve a solução encontrada pelo CPD/UnB para unificar o login de usuários com o uso de um proxy LDAP e por fim, a Subseção 3.4 descreve o esquema de autenticação e autorização proposto para os serviços REST.

3.1. Plataforma ERLANGMS

A plataforma ERLANGMS é constituído por um barramento de serviços (*Enterprise Service Bus*— ESB) multiplataforma orientado a contratos de serviços idealizado para servir de elo entre os sistemas da Universidade e a camada de serviço (tipicamente implementada usando a linguagem Java). De acordo com [Haupt et al. 2014], um barramento permite unificar o acesso aos serviços através de uma camada intermediadora entre componentes de software (denominados serviços) e as aplicações que consomem estes serviços. A implementação de um novo barramento (em vez da adoção de um barramento existente), possibilitou uma melhor compreensão do estilo arquitetural REST e o domínio

¹ Está em desenvolvimento o SDK .Net para implementação de serviços nas linguagens C# e xVB.Net.

de alguns elementos chave definidos em ERLANGMS, como a estrutura de eventos e os recursos de tolerância a falha.

A arquitetura segue o conceito de *Service Oriented Computing* (SOC), um paradigma que promove a composição de serviços *em uma rede de serviços* fracamente acoplados, com o objetivo de criar processos de negócio dinâmicos e flexíveis através da interconexão de sistemas computacionais [Haupt et al. 2014]. Dessa forma, o barramento suporta a mediação, roteamento, transformação de dados e a orquestração dos serviços. Para isso, adotou-se o estilo arquitetural REST e o formato JSON para o envio e recebimento das mensagens do cliente. Essa restrição de design teve o objetivo de facilitar a implementação do barramento e mantê-lo simples.

O esquema de comunicação da arquitetura ocorre por meio de duas vias distintas, como ilustra a Figura 1: Na primeira via, existe a comunicação do cliente para consumir algum serviço no barramento. Essa comunicação é via uma interface REST, razão pela qual o cliente (que pode ser qualquer sistema, independente da sua linguagem de programação ou plataforma) precisa suportar chamadas de serviços em REST. Na segunda via, tem a comunicação do barramento com o serviço, que está implementado em alguma linguagem de programação (Erlang, Java, etc.). Essa comunicação dá-se via sistema de mensageria disponível em Erlang que possibilita uma comunicação assíncrona com várias linguagens de programação de forma muito rápida por trafegar os dados no formato binário e com baixa latência na rede [Armstrong 2013].



Figura 1. Roteamento das mensagens na plataforma ERLANGMS.

3.2. Desenvolvimento de serviços utilizando o processo de modernização SMSOC

A introdução de um processo surgiu com a necessidade de documentar um processo de modernização para auxiliar os trabalhos de modernização. O processo SMSOC é aderente à arquitetura SOA e foi validado como resultado de um estudo de caso conduzido em uma disciplina de Pós-Graduação do Mestrado Acadêmico em Informática da UnB, através do qual foi modernizado o Sistema de Estudo SocioEconômico (SAE) que faz a gestão do processo de avaliação socioeconômica dos estudantes da UnB.

Para guiar as atividades de modernização, o processo divide-se em 4 fluxos de trabalho com atividades gerenciais e técnicas. Cada atividade apresenta um conjunto de diretrizes para a migração dos sistemas legados para um ambiente distribuído. O SMSOC documenta também a arquitetura e o design para implementação dos serviços e adota uma abordagem de desenvolvimento Domain-Driven Design (DDD), introduzido no livro *Domain-Driven Design: Tackling Complexity in the Heart of Software* para o desenvolvimento de sistemas complexos centrado no domínio do negócio e no trabalho cooperativo entre especialistas do negócio e desenvolvedores [Evans 2004].

Há vários benefícios observados com o SMSOC. Como é um processo enxuto, possibilita a sua execução até mesmo em projetos ou equipes muito pequenas. Além disso o processo apresenta diretrizes e boas práticas recomendadas além de conter um design padronizado da arquitetura dos serviços para facilitar o aprendizado por novos membros na equipe de desenvolvimento e servir como documentação da arquitetura. Outro benefício observado é quanto a facilidade de uso, principalmente da arquitetura definida no processo para o desenvolvimento dos serviços. Por exemplo, em Dezembro de 2016 houve o treinamento da equipe de desenvolvedores do setor administrativo. Após este treinamento, a equipe já implementou vários serviços solicitados pela UnB. Por outro lado, identificou-se certa resistência por desenvolvedores com mais experiência na arquitetura antiga utilizada pelo CPD/UnB em virtude dos sistemas legados trabalharem com a abordagem *TransactionScript* a qual organiza as lógica de negócio em procedimentos na camada de negócio. Uma comparação destas duas abordagens pode ser encontrada em [Agilar 2016], através do qual foi realizado uma análise destas duas abordagens para verificar qual das duas seria a ideal para o CPD/UnB.

3.3. Autenticação de usuários através um serviço proxy LDAP

Desde 2013, busca-se uma forma de unificar o acesso dos usuários aos sistemas de informação e a rede corporativa da UnB, uma vez que existem muitos silos de dados de usuários que dificultam a centralização dessas informações.

De modo a suprir esta demanda e subsidiar a concretização do acesso unificado bem como o compartilhamento de informações de usuários com as aplicações (até mesmo com sistemas não desenvolvidos pelo CPD/UnB), foi implementado um serviço proxy LDAP no barramento de serviços. O LDAP ou Lightweight Directory Access Protocol é um protocolo padrão comumente utilizado pelas aplicações para autenticar os usuários, sendo um dos protocolos de autenticação mais utilizados no ambiente da UnB.

Desde que o serviço foi colocado em produção para autenticar alguns sistemas na UnB, como é o caso do Redmine (Sistema de gerenciamento de projetos) e muito em breve o SEI 3.0 (Sistema Eletrônico de Informações do Governo Federal) que está em homologação, o proxy LDAP tem sido visto pela Direção do CPD/UnB como uma ferramenta com alto potencial para integrar não somente os sistemas mas também a rede corporativa (Webmail e UnB Wireless) da Universidade.

Note que a maior vantagem de se utilizar um serviço que entende LDAP em vez de uma instância OpenLDAP convencional por exemplo, é permitir que base de usuários possa estar em um banco de dados relacional onde tipicamente os sistemas da própria Instituição já tem acesso direto. Os demais sistemas, sites institucionais e gerenciadores de conteúdo (Joomla, Wordpress, etc.) podem consultar e autenticar os usuários através do serviço proxy LDAP da mesma forma que faria se fosse um servidor OpenLDAP.

3.4. Autenticação e Autorização de serviços REST com um serviço OAuth2

Com a modernização de sistemas através de uma abordagem SOA, torna-se importante controlar o acesso e a autorização das chamadas de serviços. Nesse sentido, está em desenvolvimento suporte nativo para autenticação utilizando o OAuth 2. O OAuth 2 é um protocolo de autorização que permite que clientes tenham acesso limitado a um serviço através de trocas de mensagens e *tokens* de acesso. Um estudo prévio foi conduzido para escolher o protocolo a ser adotado na solução ERLANGMS [Ribeiro and Canedo 2016].

Resumidamente, o suporte da autenticação utilizando o OAuth 2 foi dividido em duas partes no barramento de serviços. Na primeira parte realizou-se a implementação da geração do *token* de acordo com [Hardt 2012]. A segunda parte, que representa o desafio atual, é fazer a integração com o Sistema de Controle de Acesso (SCA) de maneira agnóstica, uma vez que o barramento pode ser utilizado por qualquer organização, sendo portanto, preferível não acoplar com os sistemas internos da UnB.

4. Conclusão

A modernização dos sistemas legados ganha cada vez mais importância na UnB. Nesse sentido, o trabalho que está sendo realizado no CPD/UnB tem como finalidade subsidiar a migração dos sistemas legados de forma sistemática e incremental, mantendo os sistemas legados em funcionamento enquanto os novos sistemas são desenvolvidos.

Um ponto importante da solução proposta foi validá-la em um estudo de caso através do qual permitiu refinar a plataforma ERLANGMS e o seu processo de modernização subjacente. Uma das vantagens observadas é permitir que tanto os sistemas novos quanto os sistemas legados possam coexistir, invocando os mesmos serviços e maximizando o compartilhamento das regras de negócio. Isso foi comprovado recentemente quando foi incluída em uma aplicação em C# uma nova funcionalidade para impressão da declaração de aluno regular disponibilizada a partir de um serviço em Java.

Referências

- Agilar, E. (2016). *Uma Abordagem Orientada a Serviços para a Modernização de Sistemas Legados*. PhD thesis, Universidade de Brasília.
- Agilar, E. A., de Almeida, R. B., and Canedo, E. D. (2016). A systematic mapping study on legacy system modernization. In *The 31st ACM/SIGAPP Symposium on Applied Computing (SAC 2016)*, pages 1–6.
- Armstrong, J. (2013). *Programming Erlang: Software for a Concurrent World*. Pragmatic Bookshelf.
- Bennett, K. (1995). Legacy systems: coping with success. *Software, IEEE*, 12(1):19–23.
- Bisbal, J., Lawless, D., Wu, B., and Grimson, J. (1999). Legacy information systems: Issues and directions. *IEEE software*, 1(5):103–111.
- Evans, E. (2004). *Domain-driven design: tackling complexity in the heart of software*. Addison-Wesley Professional.
- Fielding, R. T. (2000). *Architectural styles and the design of network-based software architectures*. PhD thesis, University of California, Irvine.
- Hardt, D. (2012). The oauth 2.0 authorization framework. In *RFC 6749*.
- Haupt, F., Karastoyanova, D., Leymann, F., and Schroth, B. (2014). A model-driven approach for rest compliant services. In *Web Services (ICWS), 2014 IEEE International Conference on*, pages 129–136. IEEE.
- Ribeiro, A. S. and Canedo, E. D. (2016). Solutions analysis of authentication and authorization for service oriented architectures. In *2016 11th Iberian Conference on Information Systems and Technologies (CISTI)*, pages 1–6.